



Runtime randomization and perturbation for virtual machines.

JAVIER CABRERA ARTEAGA

Licentiate Thesis in [Research Subject - as it is in your ISP]
School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden [2022]

TRITA-ICT XXXX:XX
ISBN XXX-XX-XXXX-XXX-X

KTH School of Information and
Communication Technology
SE-164 40 Kista
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägg-
es till offentlig granskning för avläggande av licentiatexamen i [ämne/subject]
[veckodag/weekday] den [dag/day] [månad/month] [år/2022] klockan [tid/time] i
[sal/hall], Electrum, Kungl Tekniska högskolan, Kistagången 16, Kista.

© Javier Cabrera Arteaga, [month] [2022]

Tryck: Universitetsservice US AB

Abstract

Write your abstract here...

Keywords: Keyword1, keyword2, ...

Sammanfattning

Write your Swedish summary (popular description) here...

Keywords: Keyword1, keyword2, ...

Acknowledgements

Write your professional acknowledgements here...

Acknowledgements are used to thank all persons who have helped in carrying out the research and to the research organizations/institutions and/or companies for funding the research.

Name Surname,
Place, Date

Contents

Contents	vi
List of Figures	viii
List of Tables	ix
List of Acronyms	x
1 Introduction	1
1.1 Motivation	1
1.1.1 Why variants ?	1
1.2 Contributions	1
2 Background and State of the art	2
2.1 CROW	2
3 Methodology	4
3.1 Evaluation	4
3.1.1 Corpora	4
3.1.2 Metric	4
3.1.3 Setup	5
4 Results	6
4.0.1 Example	6
4.1 Metrics	8
4.1.1 Static	8
4.1.2 Program traces and execution times	9
4.1.3 Variants preservation	9
4.2 Evaluation	10
4.2.1 Static comparison	10
4.2.2 Dynamic comparison	11
4.2.3 Preservation	11
4.3 Results	11

4.4	Results	11
4.4.1	Challenges for automatic diversification	12
4.4.2	Properties for large diversification using CROW	12
4.4.3	Variant properties	13
4.5	Conclusions	14
4.5.1	Static	14
4.5.2	Dynamic	14
4.5.3	Preservation	14
4.6	Conclusions	16
5	Variant’s application	17
5.1	Security MTD	17
5.2	Reliability (CVE + fuzz) future work	17
6	Conclusion and Future Work	18
6.1	Future Work	18
6.1.1	wasm-mutate future work	18
	Appended papers	18

List of Figures

2.1	CROW workflow to generate program variants. CROW takes C/C++ source codes or LLVM bitcodes to look for code blocks that can be replaced by semantically equivalent code and generates program variants by combining them.	2
-----	---	---

List of Tables

3.1	TODO	4
3.2	CROW tweaking for variants generation. The table is composed by the name of the corpus, the timeout parameter and the count of allowed instructions during the synthesis process.	5
4.1	Wasm engines used during the diversification assessment study. The table is composed by the name of the engine and the description of the compilation process for them.	11
4.2	General diversification results. The table is composed by the name of the corpus, the number of functions, the number of succesfully diversified functions, the number of non-diversified functions and the cumulative number of variants.	12

List of Acronyms

Wasm
DTW

WebAssembly
Dynamic Time Warping